

PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHH	HHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHHHHHHHHHHHHHHH	HHHHHHHHHHHHHHHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHHHHHHHHHHHHHHH	HHHHHHHHHHHHHHHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH

```
PPPPPPPP  AAAA  TTTTTTTTTT  PPPPPPPP  RRRRRRRR  EEEEEEEEE
PPPPPPPP  AAAA  TTTTTTTTTT  PPPPPPPP  RRRRRRRR  EEEEEEEEE
PP      PP  AA      AA  TT      PP      PP  RR      RR  EE
PP      PP  AA      AA  TT      PP      PP  RR      RR  EE
PP      PP  AA      AA  TT      PP      PP  RR      RR  EE
PP      PP  AA      AA  TT      PPPPPPPP  RRRRRRRR  EEEEEEE
PPPPPPPP  AA      AA  TT      PPPPPPPP  RRRRRRRR  EEEEEEE
PPPPPPPP  AA      AA  TT      PP      RR      RR  EE
PP      AAAA      AA  TT      PP      RR      RR  EE
PP      AAAA      AA  TT      PP      RR      RR  EE
PP      AA      AA  TT      PP      RR      RR  EE
PP      AA      AA  TT      PP      RR      RR  EEEEEEE
PP      AA      AA  TT      PP      RR      RR  EEEEEEE
```

```
RRRRRRRR  EEEEEEEEE  QQQQQQ
RRRRRRRR  EEEEEEEEE  QQQQQQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RRRRRRRR  EEEEEEE   QQ      QQ
RRRRRRRR  EEEEEEE   QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EEEEEEE  QQQQ  QQ
RR      RR  EEEEEEE  QQQQ  QQ
```

PATPRE.REQ -- REQUIRE FILE FOR PATCH

Version: 'V04-000'

```
*****
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*****
```

++

MODULE: PATPRE.REQ

FACILITY: PATCH

ABSTRACT:

REQUIRE FILE TO DEFINE LOCALLY SOME
INTERESTING GENERAL SYMBOLS, STRUCTURES AND MACROS

HISTORY:

AUTHOR: K.D. MORSE 4-OCT-77

Modified by:

V03-001 MTR0025 Mike Rhodes 8-Aug-1983
Add flags bit PAT\$\$ABSOLUTE to support patching files in
absolute mode (by virtual address). NOTE that any file may
be patched in absolute mode (eg. ISAM files etc.).
Also, bit PAT\$\$NEW VERSION has been added which is used in
conjunction with PAT\$\$ABSOLUTE to determine the mapping
performed and the generation of a new version of the patched
file or if it is patched 'in place'.

V0103 CNH0017 Chris Hume 10-Oct-1979 12:00

Added OFF bit to PAT\$GL COMFAB. Removed support for /COMMAND.
Added support for /VOLUME. (PATMAI.B32 02.27, PATSTO.B32 01.17,
PATWRT.B32 02.06, [VMSLIB]QUADEF.MAR 01.20)

MODIFICATIONS:

NO. ---	DATE ---	PROGRAMMER -----	PURPOSE -----
01	7-MAR-78	K.D. MORSE	ADD PATCH AREA STRUCTURE, PAL.
02	27-APR-78	K.D. MORSE	ADD ASSEMBLER DIRECTIVE TABLE STRUCTURE, ASD.
03	17-MAY-78	K.D. MORSE	ADD MESSAGE SEVERITY LEVELS, MSGS.
04	25-MAY-78	K.D. MORSE	ADD FORWARD REFERENCE TABLE STRUCTURE, FWR.

!++

FUNCTIONAL DESCRIPTION:

This require file defines some general, local values and macros which cannot or are best not defined globally.

!--

LITERAL

PALSK_EXP_PAREA = 1,	! Expanding patch area flag
PALSK_ADD_PAREA = 0,	! Adding new patch area flag
PATSK_USER_DEF = 3,	! Code returned by PAT\$BUILD_PATH if symbol was user-defined
PATSK_MAX_ECO = 128,	! Maximum eco level allowed
PATSK_MIN_ECO = 1,	! Minimum eco level allowed
PATSK_LENPRIV = 20,	! Length of process private ISD
PATSS_COMMAND = 0,	! Command file bit for CLI to set
PATSS_JOURNAL = 1,	! Journal file bit for CLI to set
PATSS_INPUT = 2,	! Input image file bit for CLI to set
PATSS_OUTPUT = 3,	! Output image file bit for CLI to set
PATSS_UPDATE = 4,	! Update qualifier bit for CLI to set
PATSS_VOLUME = 5,	! Volume qualifier bit for CLI to set
PATSS_ABSOLUTE = 6,	! Absolute qualifier bit for CLI
PATSS_NEW_VERSION = 7,	! New Version qualifier bit for CLI
PATSM_COMMAND = 1,	! Mask of command file CLI bit
PATSM_JOURNAL = 2,	! Mask of journal file CLI bit
PATSM_INPUT = 4,	! Mask of input image file CLI bit
PATSM_OUTPUT = 8,	! Mask of output image file CLI bit
PATSM_UPDATE = 16,	! Mask of update qualifier CLI bit
PATSM_VOLUME = 32,	! Mask of volume qualifier CLI bit
PATSM_ABSOLUTE = 64,	! Mask of absolute qualifier CLI bit
PATSM_NEW_VERSION = 128;	! Mask of new_version qualifier CLI bit

!++

Definition of image section table entries

!--

BYTEBLOCKFIELDS(ISE,	! Link to next image section entry
L_NXTISE,4,	! Start virtual address in image section
L_IMGVST,4,	! Ending virtual address in image section
L_IMGVEVD,4,	! Starting virtual address of mapped image section
L_MAPVST,4,	! Ending virtual address of mapped image section
L_MAPVEVD,4);	

!++

Definition of PATCH command text block.

!--

BYTEBLOCKFIELDS(TXT,	
L_NXTBLK,4);	! Pointer to next block

!++

Definition of patch area list entry, PAL.

!--

BYTEBLOCKFIELDS(PAL,	! Prefix name
L_FLINK,4,	! Forward link

```
L_START_ADR,4,  
L_END_ADR,4,  
L_CS_NAME,4);
```

```
! Starting patch area address  
! Ending patch area address  
! ASCII name for patch area
```

```
!++  
! Definition of Assembler Directive table structure, ASD.  
!--
```

```
BYTEBLOCKFIELDS(ASD,  
  L_PC,4,  
  L_OPINFO,4,  
  B_NUM_OPRND,1);
```

```
! Unmapped PC of assembler directive  
! Address of instruction opcode table entry for directive  
! Number of operands on directive
```

```
!++  
! Generate the data structure for the Forward Reference table, FWR$.  
!--
```

```
BYTEBLOCKFIELDS(FWR,  
  L_FLINK,4,  
  L_PC,4,  
  W_OPRNDLNG,2,  
  B_NUMBYTES,1,  
  B_NTHOPRND,1,  
  A_OPRNDADR,4,  
  A_OPINFO,4,  
  B_BUFOFF,4);
```

```
! Forward link to next entry  
! PC at which operand is to be placed  
! Length of operand string  
! Number of bytes operand will take in encoded form  
! Number of operand in instruction, i.e., nth operand  
! Address of unreduced operand ascii string  
! Index into OPINFO table for instruction's opcode  
! Offset into PAT$GL_TEMP_BUF to hold encoded operand
```

```
!++  
! Define PATCH message severity levels.  
!--
```

```
LITERAL  
MSG$K_INFO = 3,  
MSG$K_WARN = 0,  
MSG$K_SEVERE = 2,  
MSG$K_FATAL = 4,  
MSG$K_SUCCESS = 1;
```


0299

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY